PATENT
Atty. Dkt.: 11884/412001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Brunnabend, Lutz Serial No.: 10/824,437 Filed: April 15, 2004 For: CORRECTION SERVER FOR LARGE DATABASE SYSTEMS	Examiner: Michael Le Confirmation No.: 6301 Art Unit: 2163

Mail Stop Appeal Brief Patent
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

Appellants submit this appeal brief in the above-referenced application. A notice of appeal was filed on April 28, 2008. A pre-appeal brief request for review was also filed on April 28, 2008. A Notice of Panel Decision from Pre-Appeal Brief Review, mailed June 13, 2008 held that the application remains under appeal.

REAL PARTY IN INTEREST

SAP Aktiengesellschaft is the real party in interest for all issues related to this application. SAP Aktiengesellschaft owns this patent application by virtue of an assignment recorded with the Office at reel 015783, frame 0225.

RELATED APPEALS OR INTERFERENCES

There are no other appeals or interferences related to this application.

STATUS OF CLAIMS

This application contains claims 1-19, all of which stand rejected as being anticipated by the cited art. Claim 20 is cancelled. All rejections are appealed.

STATUS OF AMENDMENTS

No amendments have been filed subsequent to the final rejection in this application.

SUMMARY OF CLAIMED SUBJECT MATTER

The claimed embodiments provide systems and methods for identification of analytical results that may be rendered inconsistent due to correction of data entities upon which those results are dependent. This is accomplished through the use of a data flow manager that tracks when data entities are accessed and creates a read history relating each access to the analytical result that is dependent upon that data entity. If errors are discovered and corrected in a data entity, a correction manager creates a log of the corrected data entities. The log of corrected data entities may then be compared to the read history to determine which analytical results are dependent upon the corrected data entities, thereby revealing which analytical results are possibly inconsistent. These systems and methods provide an improved method of *tracking* accesses and error correction in a system which enables additional functionality, such as analysis based on historical data, review and filtering of data corrections, etc.

Claim 1

Independent claim 1 refers to a correction server system. (*See*, the application specification at, for example, page 2, paragraph 8, lines 1-3 and FIG. 1, generally). The correction server system includes an analyzer to calculate an analytical result using at least one data entity stored in a database. (*See*, the application specification at, for example, page 2, paragraph 12, lines 1-2 and page 4, paragraph 20, lines 1-7 and the "analyzers" and database 110 of FIG. 1). The correction server system further includes a data flow manager, responsive to read requests from agents to the database, to store a read history identifying a relationship between the data entity being read and the analytical result. (*See*, the application specification at, for example, page 2, paragraph 12, lines 1-10 and data flow manager 130 of FIG. 1). The correction server system further includes a correction server that, when corrections are made to the database, identifies corrected entities in a corrected entity log and compares the corrected entity log against the read history to identify analytical results rendered possibly inconsistent due to the correction. (*See*, the application specification at, for example, page 3, paragraphs 13-14 and elements 120 of FIG. 1).

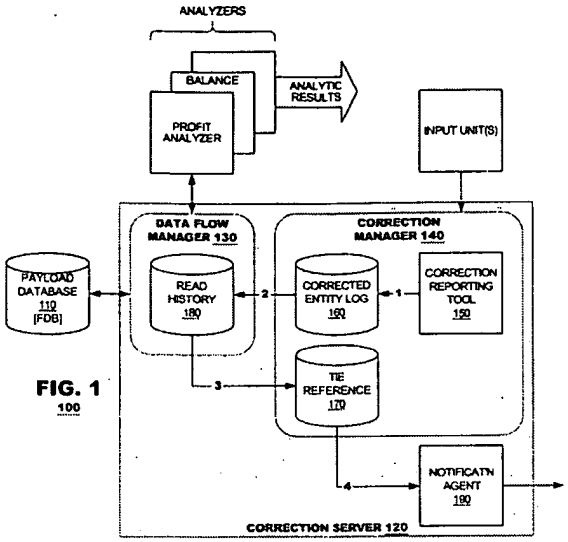

FIG. 1
100

Claim 9

Independent claim 9 refers to a computer-implemented correction management method. (*See*, the application specification at, for example, page 2, paragraph 8, lines 1-3 and FIG. 2, generally). The management method includes, responsive to a request to correct a first database entity, creating a second database entity that is a corrected copy of the first database entity. (*See*, the application specification at, for example, page 3, paragraph 15, lines 1-3 and FIG. 2, elements 210 & 220). The management method further includes storing an entry in a corrected entity log that identifies the first database entity. (*See*, the application specification at, for example, page 3, paragraph 15, lines 3-4 and FIG. 2, elements 230). The management method further includes comparing the corrected entity log entry against a read history log identifying prior accesses to the database. (*See*, the application specification at, for example, page 3, paragraph 15, lines 4-6 and FIG. 2, elements 240 & 250). The management method further includes, if the entry matches an entry from the read history log, identifying a dependent database entity from the read history log as a possibly inconsistent entity, the dependent database entity based on an analytical result calculated from the first database entity. (*See*, the application specification at, for example, page 3, paragraph 15, lines 6-7 and FIG. 2, element 260).

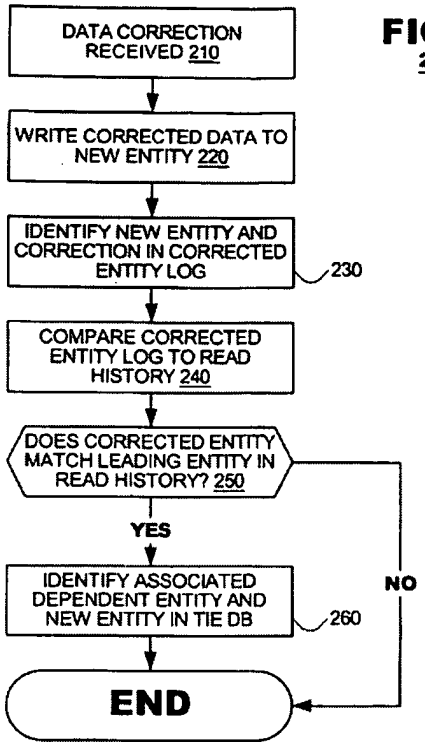

FIG. 2
200

Claim 14

Independent claim 14 refers to a computer readable medium having stored thereon program instructions. (*See*, the application specification at, for example, page 7, paragraph 29). The program instructions, when executed, cause a computer system to, responsive to a request to correct a first database entity, create a second database entity that is a corrected copy of the first database entity. (*See*, the application specification at, for example, page 3, paragraph 15, lines 1-3 and FIG. 2, elements 210 & 220). The program instructions, when executed, further cause the computer system to store an entry in a corrected entity log that identifies the first database entity. (*See*, the application specification at, for example, page 3, paragraph 15, lines 3-4 and FIG. 2, elements 230). The program instructions, when executed, further cause the computer system to compare the corrected entity log entry against a read history log identifying prior accesses to the database. (*See*, the application specification at, for example, page 3, paragraph 15, lines 4-6 and FIG. 2, elements 240 & 250). The program instructions, when executed, further cause the computer system to, if the entry matches an entry from the read history log, identify a dependent database entity from the read history log as a possibly inconsistent entity, the dependent database

entity based on an analytical result calculated from the first database entity. (*See*, the application specification at, for example, page 3, paragraph 15, lines 6-7 and FIG. 2, element 260).

Claim 19

Independent claim 19 refers to a system for identifying inconsistent data in a computer system. (*See*, the application specification at, for example, page 4, paragraph 19, and FIG. 4, generally). The system includes a first database to store data generated during operation of the computer system. (*See*, the application specification at, for example, page 2, paragraph 9, lines 2-3 and FIG. 1, element 110). The system further includes a correction manager to manage corrections performed in the system. (*See*, the application specification at, for example, page 2, paragraph 9, lines 8-11 and FIG. 1, element 140). The correction manager includes a second database to store a list of corrected data entries in the first database. (*See*, the application specification at, for example, page 2, paragraph 10, lines 2-3 and FIG. 1, element 160). The correction manager further includes a third database to store a list of uncorrected data entries identified as potentially inconsistent due to a correction performed on an entity listed in the second database. (*See*, the application specification at, for example, page 2, paragraph 10, lines 3-5 and FIG. 1, element 170). The system further includes a data flow manager to manage access to the first database, the second database, and the third database by an analyzer, the analyzer to provide analytical results calculated from data stored in the first database to an operator of the system. (*See*, the application specification at, for example, page 2, paragraph 12 and FIG. 1, elements 130 and "Analyzers").

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether the outstanding § 102 rejections to claims 1-19 in view of Burfoot U.S. Patent Publication No. 2002/0188629 should be reversed.

ARGUMENT

Burfoot does not anticipate claims 1-19 because it does not teach or suggest all of the elements of the pending claims. As explained below, in some of the Examiner's analysis, the Examiner' misinterprets the prior art. In other cases, the Examiner makes no attempt to show elements of the pending claims in violation of 37 C.F.R. §1.104(c)(2).

A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). When a claim covers several structures or compositions, either generically or as alternatives, the claim is deemed anticipated if any of the structures or compositions within the scope of the claim is known in the prior art. Brown v. 3M, 265 F.3d 1349, 1351, 60 USPQ2d 1375, 1376 (Fed. Cir. 2001). The identical invention must be shown in as complete detail as is contained in the ... claim. Richardson v. Suzuki Motor Co., 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

INDEPENDENT CLAIMS:

Claim 1 Defines Over Burfoot

Claim 1 recites, in part:

> a ***data flow manager***, responsive to read requests from agents to the database, ***to store a read history*** identifying a relationship ***between the data entity being read and the analytical result.***

The Examiner asserts that Burfoot discloses the recited data flow manager in paragraphs [0033]-[0038] and [0046] and its discussion of a "spreadsheet peer." The Examiner is incorrect.

Claim 1 requires a "data flow manager" and a "read history." The Burfoot spreadsheet peer merely reads an object from a persistence layer. (*See*, Burfoot at paragraph [0046], lines 1-4). Furthermore, Burfoot's spreadsheet peer merely uses traditional synchronization techniques to maintain consistent data:

> [T]he peer must use appropriate synchronization techniques to ensure data is not modified concurrently by multiple client applications, and **must make data commitments at appropriate times** to ensure transactional integrity.

(*See*, Burfoot at paragraph [0046] (emphasis added)). Burfoot provides no indication that the spreadsheet peer reads an operations log or otherwise stores records of client accesses and a related analytical result. Accordingly, the Examiner failed to provide any evidence that Burfoot discloses storing a "***read history*** identifying a relationship between the data entity being read and the analytical result." Burfoot's system merely prevents data inconsistency by restricting when data can be changed by the spreadsheet peer, not by tracking changes and accounting for

inconsistencies at a later point in time. This is a first basis on which claim 1 defines over the cited art.

Claim 1 further recites, in part:

> a ***corrections server*** that, when corrections are made to the database, ***identifies corrected entities in a corrected entity log*** and ***compares the corrected entity log against the read history to identify analytical results rendered possibly inconsistent*** due to the correction.

The Examiner asserts that Burfoot discloses a correction server at paragraphs [0045-46]. The Examiner is incorrect. Burfoot merely describes a DSS server that reads a DSS object, performs calculations using the DSS object and provides the resulting values to clients. There is no mention of any identification of results that are rendered possibly inconsistent by a data change. Burfoot fails to disclose, either expressly or inherently, identifying corrected entities in a corrected entity log and comparing the corrected entity log against the read history to identify analytical results rendered possibly inconsistent, because Burfoot fails to disclose a corrected entity log and a read history.

It is unsurprising that Burfoot lacks a correction server, since his system has other techniques for maintaining data consistency. In Burfoot's system, clients are not allowed to make changes that might create inconsistent data. (*See* Burfoot at paragraphs [0045-46]). Thus, Burfoot avoids data inconsistencies by **preventing** concurrent modification by multiple clients. In contrast, the claimed embodiments allows for an operator to review a change in a data entity to determine whether the change will indeed result in an inconsistency in the analytical result. (*See* Specification at paragraphs 25-28). Burfoot's system fails to describe identifying analytical results rendered possibly inconsistent due to a correction or any similar feature.

Furthermore, the Examiner provided no evidence, via citation, explanation or otherwise, that Burfoot's DSS server or calculation engine identifies corrected entities in a corrected entry log. The Examiner provided no evidence that Burfoot compares a corrected entity log to a read history. Further, the Examiner provided no evidence that Burfoot identifies possibly-inconsistent results resulting from a correction made to a database. Accordingly, the Examiner's rejection of claim 1 is improper and incomplete because the Examiner's rejection fails to meet the

minimum requirements for establishing a *prima facie* case of anticipation as set forth in 37 C.F.R. §1.104.

The rejection of claim 1 must be reversed.

Claim 9 Defines Over Burfoot

Claim 9 recites, in part:

storing an entry in ***a corrected entity log*** that identifies the first database entity,

The Examiner asserts that Burfoot discloses these features at paragraphs [0045-46]. The Examiner is incorrect. Burfoot merely describes a DSS server that reads a DSS object, performs calculations using the DSS object and provides the resulting values to clients. There is no mention of any entries being logged. In Burfoot's system, clients are not allowed to make changes that might create inconsistent data. (*See* Burfoot at paragraphs [0045-46]). Thus, Burfoot avoids data inconsistencies by **preventing** concurrent modification by multiple clients. In contrast, the claimed embodiments allows for an operator to review a change in a data entity to determine whether the change will indeed result in an inconsistency in the analytical result. (*See* Specification at paragraphs 25-28). Accordingly, Burfoot fails to disclose, either expressly or inherently, storing an entry in ***a corrected entity log*** that identifies the first database entity, because Burfoot fails to disclose or suggest logging corrected entities. This is a first basis on which claim 9 defines over the cited art.

Claim 9 further recites, in part:

> ***comparing the corrected entity log entry against a read history log*** identifying prior accesses to the database,
> if the entry matches an entry from the read history log, ***identifying a dependent database entity from the read history log as a possibly inconsistent entity***, the dependent database entity based on an analytical result calculated from the first database entity.

The Examiner asserts that Burfoot discloses these features at paragraphs [0045-46]. The Examiner is incorrect. As discussed above, Burfoot merely describes a DSS server that reads a DSS object, performs calculations using the DSS object and provides the resulting values to clients. Burfoot fails to disclose, either expressly or inherently, comparing the corrected entity log entry against a read history log and identifying a dependent database entity from the read history log as a possibly inconsistent entity.

It is unsurprising that Burfoot fails to disclose the features of the claimed embodiment, since Burfoot's system has other techniques for maintaining data consistency. In Burfoot's system, clients are not allowed to make changes that might create inconsistent data. (*See* Burfoot at paragraphs [0045-46]). Thus, Burfoot avoids data inconsistencies by **preventing** concurrent modification by multiple clients. In contrast, the claimed embodiments allows for an operator to review a change in a data entity to determine whether the change will indeed result in an inconsistency in the analytical result. (*See* Specification at paragraph 25-28). Burfoot's system fails to describe identifying a dependent database entity from the read history log as a possibly inconsistent entity.

Furthermore, the Examiner provided no evidence, via citation, explanation or otherwise, that Burfoot's DSS server or calculation engine stores and entry in a corrected entry log or a read history log. The Examiner provided no evidence that Burfoot compares a corrected entity log to a read history. Further, the Examiner provided no evidence that Burfoot identifies possibly-inconsistent results resulting from a correction made to a database. Accordingly, the Examiner's rejection of claim 9 is improper and incomplete because the Examiner's rejection fails to meet the minimum requirements for establishing a *prima facie* case of anticipation as set forth in 37 C.F.R. §1.104(c).

> Furthermore, claim 9 recites, in part:
> responsive to a request to correct a first database entity, ***creating a second database entity that is a corrected copy of the first database entity***

The Examiner failed to assert any art against this feature of claim 9. Accordingly, the Examiner's rejection of claim 9 is improper and incomplete because the Examiner's rejection fails to meet the minimum requirements for establishing a *prima facie* case of anticipation as set forth in 37 C.F.R. §1.104(b).

The rejection of claim 9 must be reversed.

Claim 14 Defines Over Burfoot

Claim 14 recites, in part:

> store an entry ***in a corrected entity log*** that identifies the first database entity

The Examiner asserts that Burfoot discloses these features at paragraphs [0045-46]. The Examiner is incorrect. Burfoot merely describes a DSS server that reads a DSS object, performs

calculations using the DSS object and provides the resulting values to clients. There is no mention of a log storing corrected entities. In Burfoot's system, clients are not allowed to make changes that might create inconsistent data. (*See* Burfoot at paragraphs [0045-46]). Thus, Burfoot avoids data inconsistencies by **preventing** concurrent modification by multiple clients. In contrast, the claimed embodiments allows for an operator to review a change in a data entity to determine whether the change will indeed result in an inconsistency in the analytical result. (*See* Specification at paragraphs 25-28). Burfoot fails to disclose, either expressly or inherently, storing an entry in ***a corrected entity log*** that identifies the first database entity. This is a first basis on which claim 14 defines over the cited art.

Claim 14 further recites, in part:

> ***compare the corrected entity log entry against a read history log*** identifying prior accesses to the database,
> if the entry matches an entry from the read history log, ***identify a dependent database entity from the read history log as a possibly inconsistent entity***, the dependent database entity based on an analytical result calculated from the first database entity.

The Examiner asserts that Burfoot discloses these features at paragraphs [0045-46]. The Examiner is incorrect. As discussed, Burfoot merely describes a DSS server that reads a DSS object and performs calculations using the DSS object and provides the resulting values to clients. Burfoot fails to disclose, either expressly or inherently, comparing the corrected entity log entry against a read history log and identifying a dependent database entity from the read history log as a possibly inconsistent entity, because Burfoot merely discusses updating a spreadsheet.

It is unsurprising that Burfoot fails to disclose the features of the claimed embodiment, since Burfoot's system has other techniques for maintaining data consistency. In Burfoot's system, clients are not allowed to make changes that might create inconsistent data. (*See* Burfoot at paragraphs [0045-46]). Thus, Burfoot avoids data inconsistencies by **preventing** concurrent modification by multiple clients. In contrast, the claimed embodiments allows for an operator to review a change in a data entity to determine whether the change will indeed result in an inconsistency in the analytical result. (*See* Specification at paragraphs 25-28). Burfoot's system fails to describe identifying a dependent database entity from the read history log as a possibly inconsistent entity.

Furthermore, the Examiner provided no evidence, via citation, explanation or otherwise, that Burfoot's DSS server or calculation engine stores and entry in a corrected entry log or a read history log. The Examiner provided no evidence that Burfoot compares a corrected entity log to a read history. Further, the Examiner provided no evidence that Burfoot identifies possibly-inconsistent results resulting from a correction made to a database. Accordingly, the Examiner's rejection of claim 9 is improper and incomplete because the Examiner's rejection fails to meet the minimum requirements for establishing a *prima facie* case of anticipation as set forth in 37 C.F.R. §1.104(c).

Furthermore, claim 14 recites, in part:

responsive to a request to correct a first database entity, ***creating a second database entity that is a corrected copy of the first database entity***

The Examiner failed to assert any art against this feature of claim 14. Accordingly, the Examiner's rejection of claim 14 is improper and incomplete because the Examiner's rejection fails to meet the minimum requirements for establishing a *prima facie* case of anticipation as set forth in 37 C.F.R. §1.104(b).

The rejection of claim 14 must be reversed.

Claim 19 Defines Over Burfoot

Claim 19 recites, in part:

a first database...; and
a correction manager...comprising:
a second database to store a list of corrected data entries in the first database; and
a third database to store a list of **uncorrected data entries** identified as potentially inconsistent **due to a correction performed on an entity listed in the second database.**

The Examiner asserts that Burfoot's web server is the recited second database, and Burfoot's persistence layer is the recited third database. This Examiner is incorrect. To anticipate the recited second database, Burfoot's web server would have to store a list of corrected data entries in the first database. However, Burfoot lacks any indication or suggestion that the web server stores data at all. The web server merely passes information between clients and the rest of the DSS system. (*See*, Burfoot at paragraph [0040]). Further, the differences between web servers and databases are well-known in the art, and are explicitly described in Burfoot at paragraphs [0032] and [0040]. Burfoot does not suggest that his web server also functions as a database,

and one of skill in the art would not reasonably interpret the described web server to be a database.

The Examiner also fails to show how Burfoot discloses the recited third database. According to the Examiner's analysis, Burfoot's persistence layer would have to store a list of uncorrected data entries identified as potentially inconsistent due to a correction performed on an entity listed in a second database (the web server, as interpreted by the Examiner). Even if the web server stores the recited list of corrected data entries, which the Appellants do not concede, there is no suggestion anywhere in Burfoot that the persistence layer stores a list of **uncorrected** entries based on the list of **corrected** entities. These features are simply absent from Burfoot's disclosure. Thus, Burfoot fails to disclose, either expressly or inherently, the claimed second and third databases as recited in the claims.

Further, to anticipate a claim a reference must disclose each and every element of the claim, **and** the elements must be arranged as required by the claim. (*See* M.P.E.P. §2131). The Examiner interprets Burfoot's business logic, web server, and persistence layer as the recited correction manager, second database, and third database, respectively. Claim 19 recites that the correction manager comprises the second and third databases. However, Burfoot's business logic does not comprise or contain the web server and persistence layer. As shown by Burfoot's Figure 1 and as described in paragraphs [0031-38], the business logic, web server, and persistence layer are **separate components** of the DSS server. There is simply no indication that Burfoot's business logic comprises a web server and a persistence layer, or that any similar configuration is possible in Burfoot's system. Thus, the reference fails to disclose all the elements, arranged in the same way, as recited in the claim.

The rejection of claim 19 must be reversed.

DEPENDENT CLAIMS

Appellants respectfully submit that the dependent claims patentably distinguish over the cited art based upon their respective dependence from an independent claim and/or for reciting patentably distinguishing features of their own.

Claim 2

Claim 2 recites:

> The correction server system of claim 1, further comprising the reading component, which ***generates a new entity from the database entity that is read*** and stores it in the database.

The Examiner asserts that Burfoot at paragraph [0046] discloses the claimed feature. The Examiner is incorrect. Burfoot at paragraph [0046] discusses a spreadsheet lookup system for finding a DSS object and further discusses a synchronization technique to "ensure data is not modified concurrently by multiple client applications." (*See*, Burfoot at paragraph [0046]). Accordingly, Burfoot fails to disclose, either expressly or inherently, generating a new entity from the database entity that is read, because as discussed, Burfoot merely describes a DSS server that reads a DSS object and performs calculations using the DSS object and provides the resulting values to clients. There is no evidence that Burfoot's DSS server generates a new entity from the database entity that is read and stores it in the database.

The rejection of claim 2 must be reversed.

Claim 3

Claim 3 recites:

> The correction server system of claim 1, wherein ***the read history log identifies leading and dependent entities***, a leading entity being a database entity that is read by a component and a dependent entity being a new object entity created from the database entity that is read.

The Examiner asserts that Burfoot at paragraphs [0011-13] and [0016-0019] discloses the claimed feature. The Examiner is incorrect. Burfoot at paragraph [0011-13] and [0016-0019] discusses a spreadsheet application and a DSS server. However, as discussed above, the Examiner failed to provide any evidence that Burfoot discloses a "read history log." Accordingly, Burfoot fails to disclose, either expressly or inherently, a read history log which identifies leading and dependent entities, because Burfoot merely describes a DSS server that reads a DSS object and performs calculations using the DSS object and provides the resulting values to clients.

The rejection of claim 3 must be reversed.

Claim 4

Claim 4 recites:

The correction server system of claim 1, ***wherein the read history stores pairs of entity identifiers***.

The Examiner asserts that Burfoot at paragraph [0097] and FIG. 3 discloses the claimed feature. The Examiner is incorrect. Burfoot at paragraph [0097] discusses that a spreadsheet object can be identified by a string. However, as discussed above, the Examiner failed to provide any evidence that Burfoot discloses a "read history," and, further, the Examiner failed to provide any evidence that pairs of entity identifies of read data are stored. Accordingly, Burfoot fails to disclose, either expressly or inherently, a read history which stores pairs of entity identifiers, because Burfoot merely describes string associated with an object on a spreadsheet.

The rejection of claim 4 must be reversed.

Claim 5

Claim 5 recites:

The correction server system of claim 1, wherein the correction server receives ***correction data*** that includes ***an identifier of a database entity being corrected, an indication of fields within the database entity that are being changed and an identification of field values that are changed***.

The Examiner asserts that Burfoot at paragraph [0097] and FIGS. 1-4 discloses the claimed feature. The Examiner is incorrect. Burfoot at paragraph [0097] discusses that a spreadsheet object can be identified by a string and that spreadsheet data can be modified. However the Examiner failed to provide any evidence that Burfoot discloses correction data that includes an identifier of a database entity being corrected, an indication of fields within the database entity that are being changed and an identification of field values that are changed. Accordingly, Burfoot fails to disclose, either expressly or inherently, that Burfoot's DSS server receives correction data that includes an identifier of a database entity being corrected, an indication of fields within the database entity that are being changed and an identification of field values that are changed, because Burfoot merely describes that a spreadsheet may be updated.

The rejection of claim 5 must be reversed.

Claim 6

Claim 6 recites:

The correction server system of claim 5, wherein the ***corrected entity log stores all the correction data*** noted in claim 5

The Examiner asserts that Burfoot at paragraph [0046] discloses the claimed feature. The Examiner is incorrect. Burfoot at paragraph [0046] describes a DSS server that reads a DSS object and performs calculations using the DSS object and provides the resulting values to clients. However the Examiner failed to provide any evidence that Burfoot discloses a corrected entity log that stores all the correction data. Accordingly, Burfoot fails to disclose, either expressly or inherently, the that corrected entity log stores all the correction data, because Burfoot at paragraph [0046] merely describes that a DSS server and there is no evidence that Burfoot's system uses a log.

The rejection of claim 6 must be reversed.

Claim 7

Claim 7 recites:

> The correction server system of claim 1, wherein the correction server further comprises a filtering agent that compares correction information to filtering criterion and ***stores the correction information in the corrected entity log only if the correction information matches the filtering criterion***

The Examiner asserts that Burfoot at paragraph [0017] and FIGS. 1-4 discloses the claimed feature. The Examiner is incorrect. Burfoot at paragraph [0017] describes FIG. 2 of Burfoot and describes how a bank may use a DSS server to analyze a portfolio. However the Examiner failed to provide any evidence that Burfoot discloses storing correction information in a corrected entity log only if the correction information matches the filtering criterion. Furthermore, the Examiner failed to provide any evidence that Burfoot discloses "filtering criterion." Accordingly, Burfoot fails to disclose, either expressly or inherently, a filtering agent that compares correction information to filtering criterion and stores the correction information in the corrected entity log only if the correction information matches the filtering criterion, because Burfoot at paragraph [0017] merely describes that a DSS server can be used by a bank to analyze a portfolio.

The rejection of claim 7 must be reversed.

Claim 8

Claim 8 recites:

The correction server system of claim 1, wherein the correction server further includes a user interface that permits ***review and display of the corrected entity log***, the user interface providing a "jump to" feature that, when activated with respect to an entry of the log causes a data entity referenced by the entry to be retrieved and displayed.

The Examiner asserts that Burfoot at paragraph [0041] discloses the claimed feature. The Examiner is incorrect. Burfoot at paragraph [0041] describes that a user can use a web browser to view and edit spreadsheet information. However the Examiner failed to provide any evidence that Burfoot discloses a corrected entity log. Accordingly, , either expressly or inherently, a user interface that permits review and display of the corrected entity log, because Burfoot at paragraph [0041] merely describes a html web browser and there is no evidence that Burfoot discloses a corrected entity log.

The rejection of claim 8 must be reversed.

Claim 10

Claim 10 recites:

The correction management method of claim 9, ***wherein the read history log stores paired leading entity identifiers and dependent entity identifiers relating to the prior accesses.***

The Examiner asserts that Burfoot at paragraph [0097] and FIG. 3 discloses the claimed feature. The Examiner is incorrect. Burfoot at paragraph [0097] discusses that a spreadsheet object can be identified by a string. However, as discussed above, the Examiner failed to provide any evidence that Burfoot discloses a "read history log," and, further, the Examiner failed to provide any evidence that paired leading entity identifiers and dependent entity identifiers of read data are stored. Accordingly, Burfoot fails to disclose, either expressly or inherently, a read history log stores paired leading entity identifiers and dependent entity identifiers relating to the prior accesses, because there is no evidence that Burfoot discloses a read history log.

The rejection of claim 10 must be reversed.

Claim 11

Claim 11 recites:

The correction management method of claim 10, wherein the comparison is made ***between an entity identifier from the corrected entity log and the leading entity identifier from the read history log.***

The Examiner asserts that Burfoot at paragraphs [0028-36] and FIG. 3 discloses the claimed feature. The Examiner is incorrect. The Examiner refers to a "DSS server and external programs that interact with the DSS server" in asserting the rejection. However, as discussed above, the Examiner failed to provide any evidence that Burfoot discloses a "read history log" or a "corrected entity log." Accordingly, Burfoot cannot disclose, either expressly or inherently, that a comparison is made between an entity identifier from the corrected entity log and the leading entity identifier from the read history log.

The rejection of claim 11 must be reversed.

Claim 12

Claim 12 recites:

> The correction management method of claim 9, wherein the correction includes an ***entity identifier of the first database entity and an indication of fields within the first database entity being corrected.***

The Examiner asserts that Burfoot at paragraphs [0011-13] and [0016-0019] discloses the claimed feature. The Examiner is incorrect. Burfoot at paragraph [0011-13] and [0016-0019] discusses a spreadsheet application and a DSS server. However there is no evidence that Burfoot discloses a correction which includes an entity identifier of the first database entity and an indication of fields within the first database entity being corrected. Accordingly, Appellants respectfully submit that Burfoot fails to disclose, either expressly or inherently, the features of claim 12.

The rejection of claim 12 must be reversed.

Claim 13

Claim 13 recites:

> The correction management method of claim 9, further comprising comparing the correction request ***to filtering criteria*** and performing the storing and comparing ***unless the correction request does not satisfy the filtering criteria.***

The Examiner asserts that Burfoot at paragraphs [0046] and [0097] discloses the claimed feature. The Examiner is incorrect. As discussed above, Burfoot at paragraphs [0046] and [0097] discuss a DSS server updating a spreadsheet. However, there is no evidence that Burfoot discloses a "filtering criteria" or performing the storing and comparing unless the correction request does

not satisfy the filtering criteria. Accordingly, Burfoot fails to disclose, either expressly or inherently, comparing the correction request to filtering criteria and performing the storing and comparing unless the correction request does not satisfy the filtering criteria.

The rejection of claim 13 must be reversed.

Claim 15

Claim 15 recites:

> The medium of claim 14, wherein the ***read history log stores paired leading entity identifiers and dependent entity identifiers*** relating to the prior accesses.

The Examiner asserts that Burfoot at paragraph [0097] and FIG. 3 discloses the claimed feature. The Examiner is incorrect. Burfoot at paragraph [0097] discusses that a spreadsheet object can be identified by a string. However, as discussed above, the Examiner failed to provide any evidence that Burfoot discloses a "read history log," and, further, the Examiner failed to provide any evidence that paired leading entity identifiers and dependent entity identifiers of read data are stored. Accordingly, Burfoot cannot disclose, either expressly or inherently, that a read history log stores paired leading entity identifiers and dependent entity identifiers relating to the prior accesses.

The rejection of claim 15 must be reversed.

Claim 16

Claim 16 recites:

> The medium of claim 15, wherein the comparison is made between ***an entity identifier from the corrected entity log and the leading entity identifier from the read history log.***

The Examiner asserts that Burfoot at paragraphs [0028-36] and FIG. 3 discloses the claimed feature. The Examiner is incorrect. The Examiner refers to a "DSS server and external programs that interact with the DSS server" in asserting the rejection. However, as discussed above, the Examiner failed to provide any evidence that Burfoot discloses a "read history log" or a "corrected entity log." Accordingly, Burfoot cannot disclose, either expressly or inherently, the a comparison is made between an entity identifier from the corrected entity log and the leading entity identifier from the read history log.

The rejection of claim 16 must be reversed.

Claim 17

Claim 17 recites:

The medium of claim 14, wherein the ***correction request includes an entity identifier of the first database entity and an indication of fields within the first database entity being corrected.***

The Examiner asserts that Burfoot at paragraphs [0011-13] and [0016-0019] discloses the claimed feature. The Examiner is incorrect. Burfoot at paragraph [0011-13] and [0016-0019] discusses a spreadsheet application and a DSS server. However the Examiner failed to provide any evidence that Burfoot discloses a correction request which includes an entity identifier of the first database entity and an indication of fields within the first database entity being corrected. Accordingly, Burfoot fails to disclose, either expressly or inherently, the a correction request includes an entity identifier of the first database entity and an indication of fields within the first database entity being corrected.

The rejection of claim 17 must be reversed.

Claim 18

Claim 18 recites:

The medium of claim 14, further comprising comparing the correction request to ***filtering criteria*** and ***performing the storing and comparing unless the correction request does not satisfy the filtering criteria.***

The Examiner asserts that Burfoot at paragraphs [0046] and [0097] discloses the claimed feature. The Examiner is incorrect. As discussed above, Burfoot at paragraphs [0046] and [0097] discuss a DSS server updating a spreadsheet. However, the Examiner failed to provide any evidence that Burfoot discloses a "filtering criteria" or performing the storing and comparing unless the correction request does not satisfy the filtering criteria. Accordingly, Appellants respectfully submit that Burfoot fails to disclose, either expressly or inherently, comparing the correction request to filtering criteria and performing the storing and comparing unless the correction request does not satisfy the filtering criteria.

The rejection of claim 18 must be reversed.

CONCLUSION

Applicant respectfully requests reversal of the obviousness rejection to claims 1-19. These claims are allowable over the cited art.

The Office is hereby authorized to charge any additional fees or credit any overpayments under 37 C.F.R. § 1.16 or § 1.17 to Deposit Account No. 11-0600.

Respectfully submitted,

Date: July 31, 2008

/Matthew H. Polson/
Matthew H. Polson
Registration No. 58,841

KENYON & KENYON LLP
1500 K Street, N.W.
Washington, D.C. 20005
Ph.: (202) 220-4200
Fax.: (202) 220-4201

APPENDIX

1. (Previously Presented) A correction server system, comprising:

an analyzer to calculate an analytical result using at least one data entity stored in a database;

a data flow manager, responsive to read requests from agents to the database, to store a read history identifying a relationship between the data entity being read and the analytical result, and

a correction server that, when corrections are made to the database, identifies corrected entities in a corrected entity log and compares the corrected entity log against the read history to identify analytical results rendered possibly inconsistent due to the correction.

2. (Original) The correction server system of claim 1, further comprising the reading component, which generates a new entity from the database entity that is read and stores it in the database.

3. (Previously Presented) The correction server system of claim 1, wherein the read history log identifies leading and dependent entities, a leading entity being a database entity that is read by a component and a dependent entity being a new object entity created from the database entity that is read.

4. (Original) The correction server system of claim 1, wherein the read history stores pairs of entity identifiers.

5. (Previously Presented) The correction server system of claim 1, wherein the correction server receives correction data that includes an identifier of a database entity being corrected, an indication of fields within the database entity that are being changed and an identification of field values that are changed.

6. (Original) The correction server system of claim 5, wherein the corrected entity log stores all the correction data noted in claim 5.

7. (Original) The correction server system of claim 1, wherein the correction server further comprises a filtering agent that compares correction information to filtering criterion and

stores the correction information in the corrected entity log only if the correction information matches the filtering criterion.

8. (Original) The correction server system of claim 1, wherein the correction server further includes a user interface that permits review and display of the corrected entity log, the user interface providing a "jump to" feature that, when activated with respect to an entry of the log causes a data entity referenced by the entry to be retrieved and displayed.

9. (Previously Presented) A computer-implemented correction management method, comprising:

responsive to a request to correct a first database entity, creating a second database entity that is a corrected copy of the first database entity,

storing an entry in a corrected entity log that identifies the first database entity,

comparing the corrected entity log entry against a read history log identifying prior accesses to the database,

if the entry matches an entry from the read history log, identifying a dependent database entity from the read history log as a possibly inconsistent entity, the dependent database entity based on an analytical result calculated from the first database entity.

10. (Original) The correction management method of claim 9, wherein the read history log stores paired leading entity identifiers and dependent entity identifiers relating to the prior accesses.

11. (Original) The correction management method of claim 10, wherein the comparison is made between an entity identifier from the corrected entity log and the leading entity identifier from the read history log.

12. (Original) The correction management method of claim 9, wherein the correction includes an entity identifier of the first database entity and an indication of fields within the first database entity being corrected.

13. (Original) The correction management method of claim 9, further comprising comparing the correction request to filtering criteria and performing the storing and comparing unless the correction request does not satisfy the filtering criteria.

14. (Previously Presented) Computer readable medium having stored thereon program instructions that, when executed, cause a computer system to:

responsive to a request to correct a first database entity, create a second database entity that is a corrected copy of the first database entity,

store an entry in a corrected entity log that identifies the first database entity,

compare the corrected entity log entry against a read history log identifying prior accesses to the database,

if the entry matches an entry from the read history log, identify a dependent database entity from the read history log as a possibly inconsistent entity, the dependent database entity based on an analytical result calculated from the first database entity.

15. (Original) The medium of claim 14, wherein the read history log stores paired leading entity identifiers and dependent entity identifiers relating to the prior accesses.

16. (Original) The medium of claim 15, wherein the comparison is made between an entity identifier from the corrected entity log and the leading entity identifier from the read history log.

17. (Original) The medium of claim 14, wherein the correction request includes an entity identifier of the first database entity and an indication of fields within the first database entity being corrected.

18. (Original) The medium of claim 14, further comprising comparing the correction request to filtering criteria and performing the storing and comparing unless the correction request does not satisfy the filtering criteria.

19. (Previously Presented) A system for identifying inconsistent data in a computer system, comprising:

a first database to store data generated during operation of the computer system;

a correction manager to manage corrections performed in the system, the correction manager further comprising:

a second database to store a list of corrected data entries in the first database; and

a third database to store a list of uncorrected data entries identified as potentially inconsistent due to a correction performed on an entity listed in the second database; and

a data flow manager to manage access to the first database, the second database, and the third database by an analyzer, the analyzer to provide analytical results calculated from data stored in the first database to an operator of the system.

20. (Canceled).

Evidence Appendix

None

Related Proceedings Appendix

None